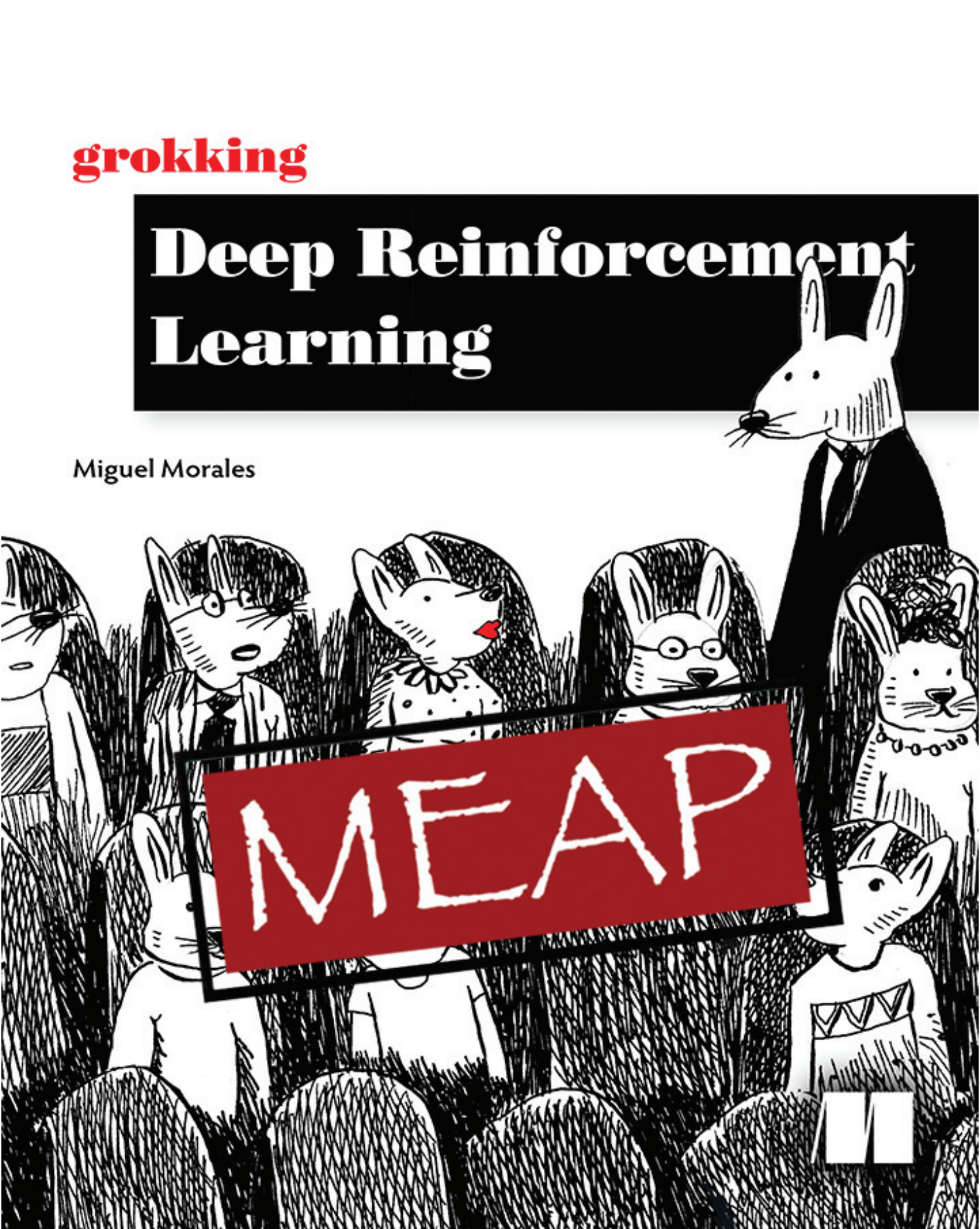
grokking

Deep Reinforcement Learning

Miguel Morales



MEAP Edition
Manning Early Access Program
Grokking Deep Reinforcement Learning
Version 2

For more information on this and other Manning titles go to
www.manning.com

welcome

Thanks for purchasing the MEAP for *Grokking Deep Reinforcement Learning*. My vision is that by buying this book, you will not only learn deep reinforcement learning but also become an active contributor to the field. Deep reinforcement learning has the potential to revolutionize the world as we know it. By removing humans from decision-making processes, we set ourselves up to succeed. Humans can't match the stamina and work ethic of a computer; we also have biases that make us less than perfect. Imagine how many decision-making applications could be improved with the objectivity and optimal decision making of a machine—healthcare, education, finance, defense, robotics, etc. Think of any process in which a human repeatedly makes decisions; deep reinforcement learning can help in most of them. Deep reinforcement learning can do great things as it is today, but the field is still not perfect. That should excite you, because it means we need people with the interest and skills to push the boundaries of this field forward. We are lucky to be part of this world at this point, and we should take advantage of it and make history. Are you up for the challenge?

I've been involved in Reinforcement Learning for a few years now. I first studied the topic in a course at Georgia Tech: Reinforcement Learning and Decision Making, which was co-taught by Drs. Charles Isbell and Michael Littman. It was inspiring to hear from top researchers in the field, interact with them daily, and listen to their perspectives. The following semester, I became a Teaching Assistant for the course and never looked back. Today, I'm an Instructional Associate at Georgia Tech and continue to help with the class daily. I've been privileged to interact with top researchers in the field and with hundreds of students, and I've become a bridge between the experts and the students for almost two years now. I understand the gaps in knowledge, the topics that are often the source of confusion, the students' interests, the foundational knowledge that is classic yet necessary, the classical papers that can be skipped, and many other things that put me in a position to write this book. In addition to teaching at Georgia Tech, I work full-time for Lockheed Martin, Missile and Fire Control - Autonomous Systems. We do top autonomy work, part of which involves the use of autonomous decision-making such as in deep reinforcement learning. I felt inspired to take my passion for both teaching and deep reinforcement learning to the next level by making this field available to anyone who is willing to put in the work.

I partnered with Manning to deliver a great book to you. Our goal is to help readers understand how deep learning makes reinforcement learning a more effective approach. In the first part of the book, we will dive into the foundational knowledge specific to reinforcement learning. Here you'll gain the necessary expertise to solve more complex decision-making problems. In the second part, I'll teach you to use deep learning techniques to solve massive, complex reinforcement learning problems. We will dive into the top deep reinforcement learning algorithms and dissect them one at a time. Finally, in

the third part, we will look at advanced applications of these techniques. We will put everything together then and help you see the potential of this technology.

Again, it is an honor to have you with me; I hope that I can inspire you to give your best and apply the knowledge you will obtain in this book to solve complex decision-making problems and make this a better place. Humans may be sub-optimal decision makers, but buying this book was without a doubt the right thing to do. Let's get working.

—Miguel Morales

brief contents

Introduction to Deep Reinforcement Learning 1

IN THIS CHAPTER

You'll learn what deep reinforcement learning is and where it comes from laying the foundation for later chapters.

You'll understand how deep reinforcement learning is embedded in a larger field of related approaches and how these relationships influence this field.

You'll recognize how this approach is different to other machine learning approaches and why it is important.

You'll identify what deep reinforcement learning can accomplish for a variety of problems.

"I visualize a time when we will be to robots what dogs are to humans, and I'm rooting for the machines."

— Claude Shannon
Father of the Information Age and
the field of Artificial Intelligence

Humans and animals naturally pursue feelings of happiness. From picking our daily meals to going after long-term goals, every action we choose is derived from our drive to experience rewarding moments in life. Whether these moments are self-centered pleasures or the more altruistic of goals, they are still our perception of how important and rewarding they are. And this is, to some extent, our reason for living.

Our ability to achieve these rewarding moments, especially those that take time to come to fruition, seems to be correlated with intelligence; intelligence being defined as the ability to *acquire* and *apply* knowledge and skills. People that are deemed by society as intelligent display an ability to balance immediate and long-term goals. Goals that take longer to materialize are normally the hardest to achieve, and it is those who are able to withstand the challenges along the way that are the exception, the leaders, the intellectuals of society.

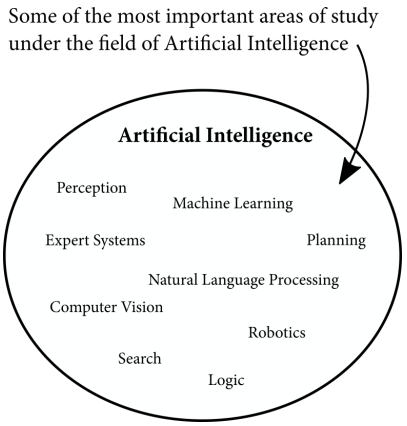
In this book, you will learn a computer science approach known as deep reinforcement learning. Deep reinforcement learning studies the design and creation of machine agents that can mimic human intelligence by receiving a stream of data, acting in the environment around them and learning from trial and error.

What is Deep Reinforcement Learning?

Deep reinforcement learning is part of a broader field—artificial intelligence—that studies the creation of intelligent computer programs. Before we dive into the specifics of deep reinforcement learning, let's make sure you understand how it fits into this branch of computer science.

Artificial Intelligence

Artificial intelligence (AI) is a branch of computer science that studies computer programs capable of displaying intelligence. "Intelligence" may seem like a broad term, but traditionally, any piece of software that displays cognitive abilities such as perception, search, planning, and learning is considered part AI. Some examples



of functionality produced by AI software are:

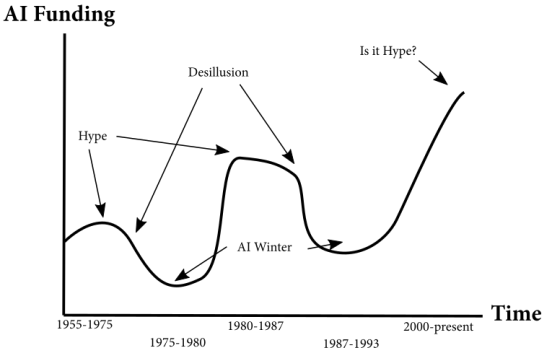
- The pages to be most likely returned by your search engine of choice.
- The route produced by your GPS app.
- The voice recognition and the artificial voice of your phone assistant app.
- The list of items related to your most recent purchase at your e-commerce site of choice.
- The follow-me feature on some of the most popular brands of drones.

Humans have been intrigued by the possibility of intelligent creatures other than ourselves since antiquity, but Alan Turing's work in the 1940s and '50s paved the way for modern computer science and AI by laying down important theoretical foundations that later scientists leveraged. The most well known of these is the **Turing Test**, which proposes a standard for measuring machine intelligence: if a human interrogator is unable to distinguish a machine from another human on a chat Q&A session, then the computer is said to count as intelligent. Though rudimentary, the Turing Test allowed generations to wonder about the possibilities of creating intelligent machines by setting a goal that researchers could pursue.

Also starting in the 1950s, an influential AI researcher named John McCarthy made several important contributions to the field. To name a few, McCarthy is credited with coining the term "artificial intelligence" in 1955, leading the first AI conference in 1956, inventing the Lisp programming language in 1958, cofounding the MIT AI Lab in 1959, and contributing major papers on AI over several decades.

All the work and progress at the time created a great deal of excitement, but there were major setbacks. Prominent researchers bet that we would be able to create an agent with human-like intelligence within just a few years, but this was an overly optimistic expectation. To make things worse, a well-known researcher named James Lighthill compiled a report criticizing the state of academic research in

Beyond actual numbers, AI has followed a pattern of hype and desillusion for years. What does the future hold?



artificial intelligence. These things contributed to a long period of reduced funding and interest in artificial intelligence research known as the first AI winter. The field has continued this pattern throughout the years: Researchers make progress, then overestimate and overcommit to results and miss deadlines, and this leads the government and industry partners to reduce or altogether cut off research funding. The recent hype around AI might lead us to worry about another winter, but the success of AI in recent years seems different.

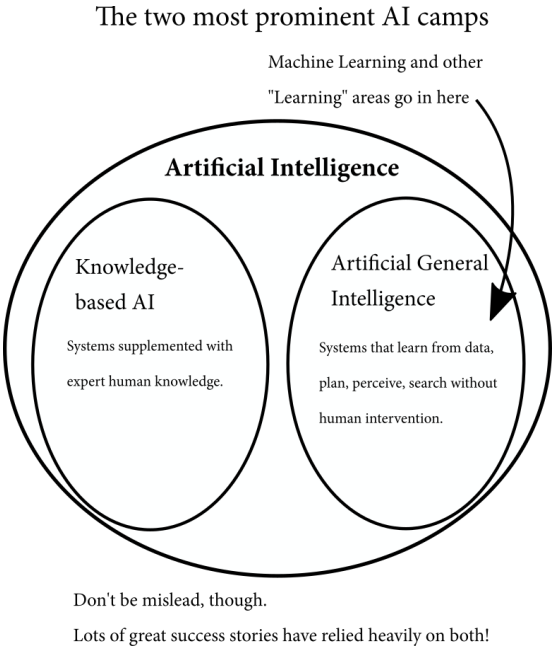
Today, the most powerful companies in the world make the largest investments to AI research. Companies like Google, Facebook, Microsoft, Amazon and Apple have invested in AI research since their founding and have become highly profitable thanks, in part, to the AI systems they've developed and acquired. Their large and steady investments have created the perfect environment for the current pace of AI research. Current researchers have the best computing power available and large amounts of data for their research, and teams of top researchers are working together on the same problems, in the same location, at the same time. Current AI research has become more stable and more productive. We have been witnessing one AI success after another, and it is not likely to stop.

Regardless of what happens in the next decades, artificial intelligence will be here for the long haul. For many, artificial intelligence is not only about creating intelligent computer software; it also addresses a more philosophical need to understand ourselves, our existence—how we think and why we do things the way we do.

Machine Learning

There are two prominent camps of artificial intelligence, one side of the spectrum a pragmatic approach, the other side a purist one. The pragmatic approaches were commonly referred to as *strong* AI because it usually performed better than the purist, *weak* AI, approaches in real-world problems. On the pragmatic side, we have **knowledge-based AI** (KBAI). KBAI consists of computer scientists who study and create AI systems supplemented with human expert knowledge. The goal of this approach is to specialize, to create expert, knowledge-based systems. It is considered strong despite the injection of human expertise. On the other side of the spectrum, we have **artificial general intelligence** (AGI). The goal of AGI, a more purist approach to creating intelligent systems, is to create an intelligent agent that can learn on its own and from scratch on a wide-ranging set of tasks. In the past, these meth-

ods were limited by low computing power and data availability, so AGI was known as weak AI. But, as technology progresses, human expertise becomes the bottleneck and the purist methods previously seen as weak become better and stronger. Deep reinforcement learning, as you will see soon, is part of the AGI side and in this book we will devote no time to KBAI.



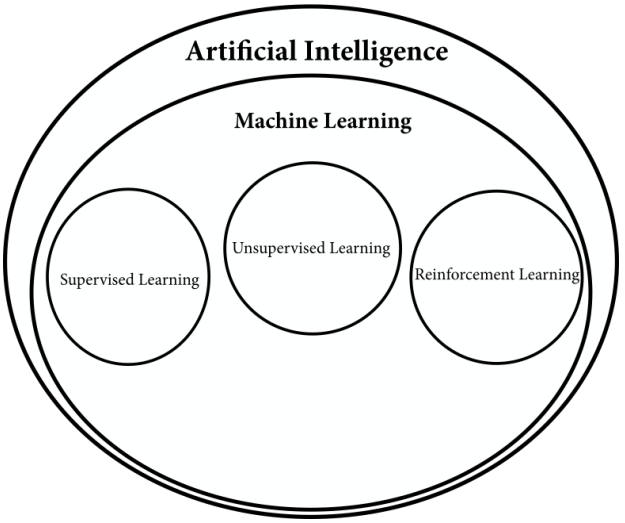
Machine learning (ML) is part of the purist AI camp as it is about learning from data. There are three main ML branches: supervised learning, unsupervised learning, and reinforcement learning. **Supervised learning** (SL) is the task of learning from labeled data. In SL, a human decides which data to collect and how to label it. A basic example of SL would be a program that can identify pictures of cats. In SL, the model learns to generalize to unseen samples. For example: a human would collect images with and without cats, label those images as such, and then train a model to classify the images as having cats or not. The trained model would then be able to classify new images as having cats or not.

Unsupervised learning (UL) is the task of learning from unlabeled data. Even though data no longer needs labeling, the methods used by the computer to gather data still need to be designed by a human. The goal in UL is to group data into

meaningful clusters. For example: a human would collect data on customers and then train a model to group those customers so that, by known what some of those customers buy, we can offer products to other customers based on common customer traits. We would not tell the machine learning model what relevant traits to use for grouping; this is what the model would learn.

Reinforcement learning (RL) is the task of learning through interaction. In this type of task, no human labels data and no human collects or designs the collection of data. These machine learning algorithms can be thought of as agents because of the need for interaction. The agents need to learn to perform a specific task, like in other machine learning paradigms. They also need to collect the most relevant data. Very often, in RL, you must provide a reward signal. This signal is fundamentally different from the labels in supervised learning. In RL, agents receive reward signals for achieving a goal and not for specific agent behaviors. Additionally, this signal is related to an obviously desired state like winning a game, reaching an objective or location, and so on, which means that humans do not need to intervene by labeling millions of samples.

Main branches of machine learning



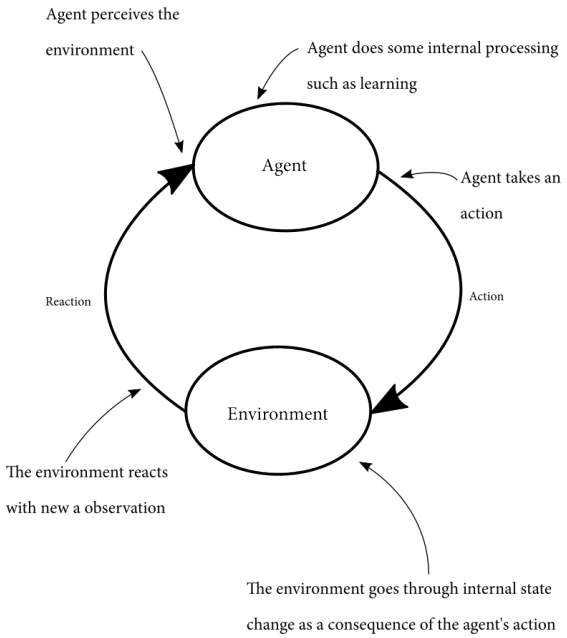
These types of machine learning tasks are all important, and they are not mutually exclusive. In fact, the best examples of artificial general intelligence are those that combine different machine learning techniques.

Reinforcement Learning

The fields of mathematics, engineering, psychology, economics, neuroscience, computer science are all interested in the problem of optimal action selection—how to come up with action policies for complex systems? Reinforcement learning is the manifestation of this need. RL can be thought of as computational behaviorism. Behaviorism is a psychological approach that studies human and animal behavior and states that interaction with the environment is the source of all behavior. The environment tries to reinforce or discourage certain behavior, and it is ultimately the human or animal that decides what action is best under specific circumstances. There is no formal connection between RL and behaviorism, but human and animal behavior can help us gain an intuitive understanding of reinforcement learning.

Think of how you would train a dog to sit. When the dog properly sits on command, you give him a treat. You know your dog likes treats, and he would probably want to know how to get more of them. He interacts with you by trying different actions: he might run, bark, stay, and many other things before sitting. Imagine how complicated it is for your dog to learn what action led to the treat. Was it the running at first? Was it giving up? Your dog needs to look back in time and assign credit to possible actions that led to the treat. It also needs to explore. Not only that, but it needs to learn to balance exploring new actions and retrying the actions that have worked best so far.

The reinforcement learning interaction cycle

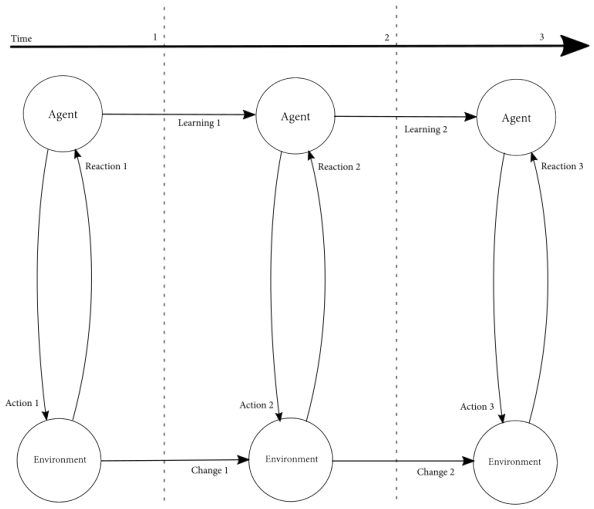


Reinforcement learning is an interaction cycle between a controller, known as the agent, and a system, known as the environment. In the dog training example, your dog is an agent and you are your dog's environment. The cycle begins with the agent observing the environment. The agent does some internal

processing of the observation like learning or memorizing. The agent then takes an action that will affect the environment in some way. Usually, the action affects the environment in two different ways: First, there is an internal change in the environment that the agent may or may not be able to see. This internal change could have delayed consequences that only manifest in future interactions. Second, the environment reacts externally, in a way that the agent can see. The environment's response could be a direct consequence of the last agent's action, but it could also be related to some earlier interaction. The whole cycle—observation, processing, action, responses from the environment—then repeats.

RL has three distinct characteristics worth highlighting. First, RL is concerned with *sequential decision-making*. In cases in which there are sequences of decisions, as opposed to single-shot decisions, your decisions may not affect only the immediate feedback signal you get, but may influence all possible future feedback signals. Being able to affect the environment complicates decision-making because we can't simply choose the action with the highest immediate reward for the current system state; instead, our decision-making must also account for the change to the environment that we will bring about.

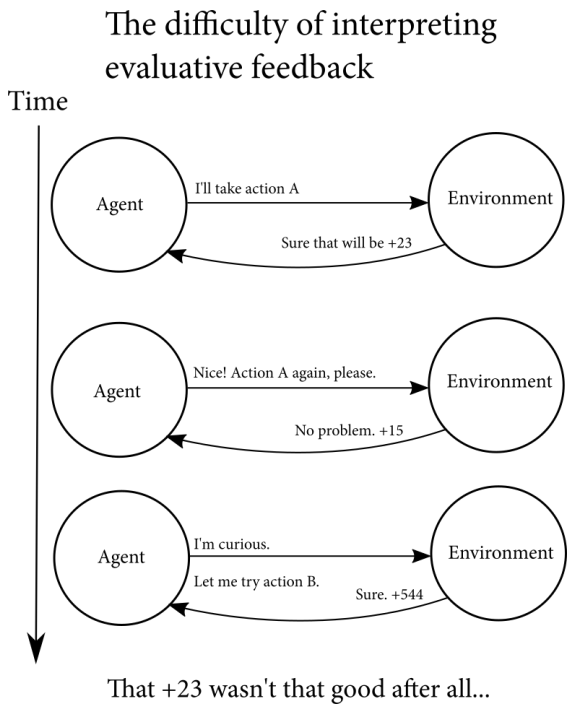
A sequential decision-making problem



The environment's reaction is an immediate feedback and might seem good at first, but the change in the environment will influence all future reactions the agent is able to perceive.

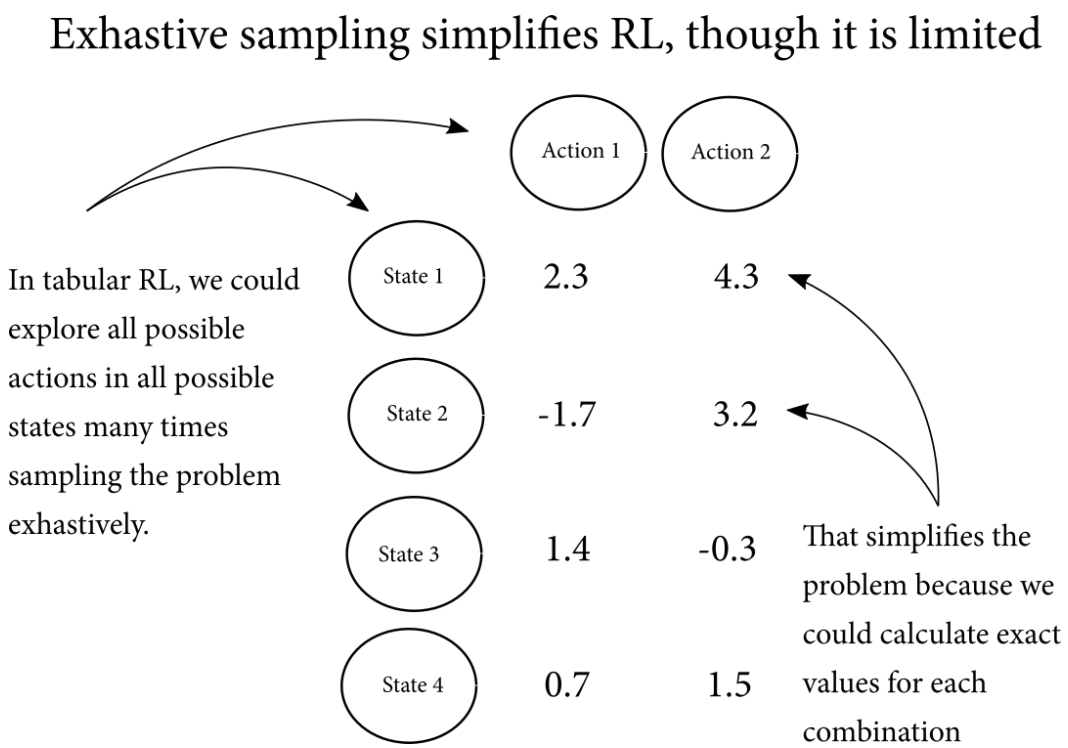
For example: deforestation is one of many sequential decision-making problems. Even though logging operations provide us with wood and paper products that we dearly need, removing trees also affects the environment in ways that could be harmful, even to ourselves. So, we cannot simply think of the action that maximizes our immediate profit, but we must think of how we will affect the environment and how that will, in turn, affect us in the future. In other machine learning paradigms like SL, this sequential nature does not exist. The problems are one-shot or single-shot problems. For example, identifying a cat in an image will not change what you will see in a next image.

The second distinct characteristic of RL is the use of *evaluative feedback*, rather than supervised. Supervised feedback, as is used in SL, is simple and straightforward. You either classified that image correctly or you didn't. Evaluative feedback, on the other hand, is more complex to interpret and categorize as good or bad, correct or incorrect. For example, if you get offered a salary of $60k, is that good or bad? What if you learn one of your future coworkers makes $75k? How good is the offer now? What if all your other future coworkers make $30k? Not so bad anymore, right?



See why evaluative feedback is so complicated? You must first gather data and then make comparisons. But there could always be some data that you haven't seen yet that would change the meaning of your current data.

RL third distinct characteristic is that it relies on an *exhaustive sampling* of the environment. In "tabular" reinforcement learning, you typically fit all possible environment state and action combinations on a table, then simulate millions of interactions with the environment, therefore exposing all possible situations you could ever encounter in that environment. Supervised learning, on the other hand uses sampled feedback. That is, you do not see all possible situations. Instead, the model is trained to generalize to unseen possibilities.

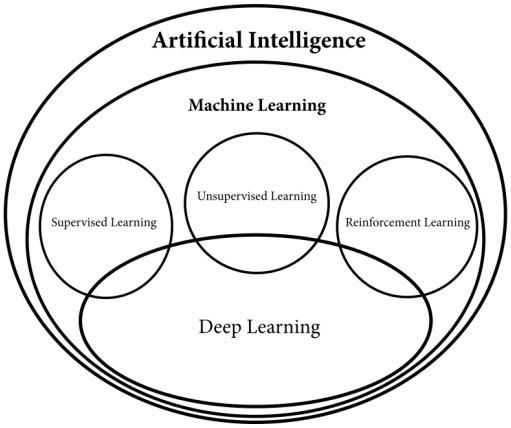


Most of the theory of reinforcement learning exists only for exhaustive sampling cases, but the union of reinforcement learning and supervised learning allows us to build reinforcement learning agents capable of solving complex problems in which exhaustive sampling is inefficient or even impossible.

Deep Learning

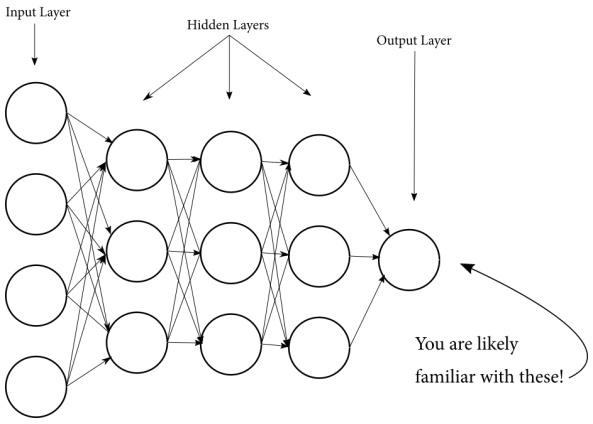
You may think that deep learning is this *new* hot technology that is allowing us to talk about self-driving cars and virtual reality. This is partially true, but deep learning is an approach to machine learning that refers to the training of expressive multi-layer models, most often artificial neural networks. And artificial neural networks (NN) have been around for decades. What makes deep learning special is that, in recent years, we have been able to go from a hard limit of three layers to networks with a double digit number of layers.

Deep learning improving all areas of Machine Learning



In **deep learning** (DL), each layer learns an abstraction of the next layer. For example, if you are training a convolutional neural network (CCN) on images of animals, the ouput layer would learn to classify the animals and each preceeding layer would learn to abstract the images into more general features. To think about it backward is not the most intuitive way, but it is important to know why I describe it this way.

A simple feedforward neural network

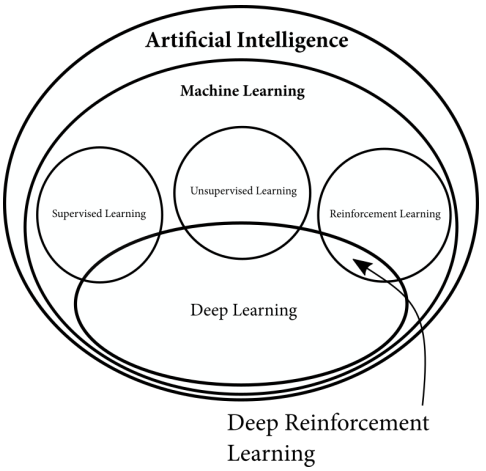


The most popular algorithm used to trained NN is **backpropagation** and it works by propagating the partial computation of the gradient from one layer backwards to compute the gradient of the previous layer. It works from output to input. However, it is arguably more intuitive to think of NN the other way around, from the input to the output layers. This other way, the first layer would learn primitive geometric features such as points, lines and curves. The next layer would learn basic geometric figures such as circles, squares and other shapes. The next layer would learn features more specific to the animals that it trained on such as ears, noses, eyes and so on. The final layer at the top would learn to classify the different animals. Because of the 3-layer limit we previously had, the number of abstractions that we could perform was not that great. Instead, experts had to create relevant abstractions by hand, so they could then train a simple neural network model on these feature vectors.

However, in 2006 a group of Canadian researchers introduced algorithms capable of breaking this limit and applied these new techniques to recognition tasks—first, handwritten digits and later, speech. The results they achieved shocked the machine learning community and revolutionized the field of AI. Every challenge that had previously been attempted using hand-crafted features for training became a great opportunity for researchers to go back and obtain groundbreaking results just by switching to DL. As a result, the deep learning community has been influencing every branch of ML and affecting industry after industry and creating the current state of research.

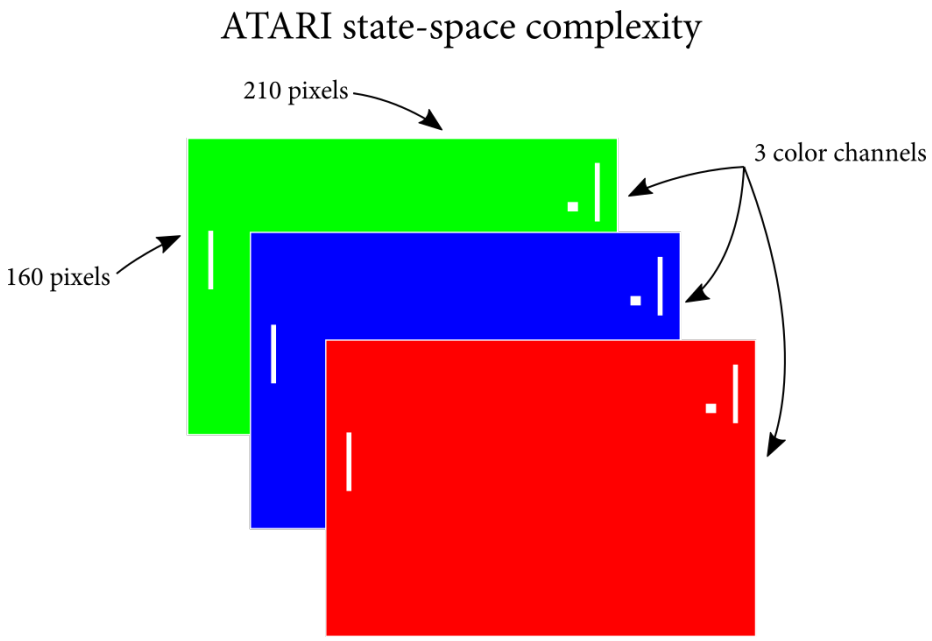
Deep Reinforcement Learning

So how is deep reinforcement learning different than reinforcement learning? As you might have guessed correctly, it's the deep part. **Deep reinforcement Learning** (DRL) is simply the use of multiple layers of powerful function approximators to solve complex sequential decision-making problems.



Imagine you want to train an agent to play ATARI games straight from the images the console outputs. ATARI games

output high dimensional data; images of 210 by 160 pixels with 3-channel video at 60 times per second. Still, this representation does not cover motion. For that, we have to use the last couple of frames to determine things like velocity of the ball and so on.

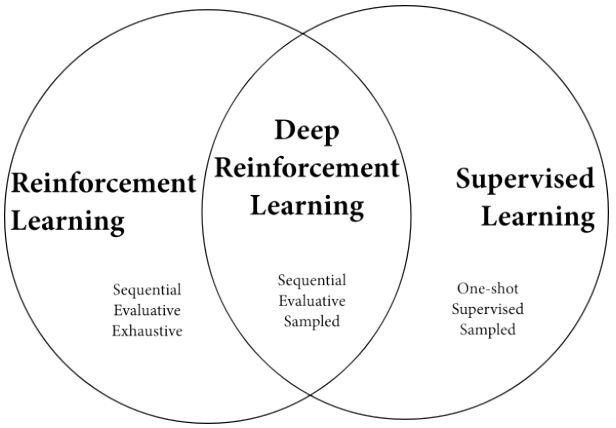


That is a very large state-space. If you compare it with the grid-world games RL is commonly associated with; you quickly sense the gap. Moreover, robotic control problems have continuous state spaces, and games like Go have more states than the number of atoms in the universe. So surely, even if you could engineer a way to store a table with all the values you could encounter in these environments, it would be very inefficient to learn this way. Instead, we should leverage generalization techniques to find approximate solutions. Given that it is very difficult, sometimes even impossible, to sample the state-space exhaustively, we need to turn towards the function approximation methods found in other areas of machine learning.

One way to create a DRL agent that could learn to play ATARI, then, is by creating a deep learning model, a convolutional neural network, to map a set of 4 consecutive images of the video stream to each of the possible actions in the ATARI game platform. The output nodes correspond to the predicted values of the individual

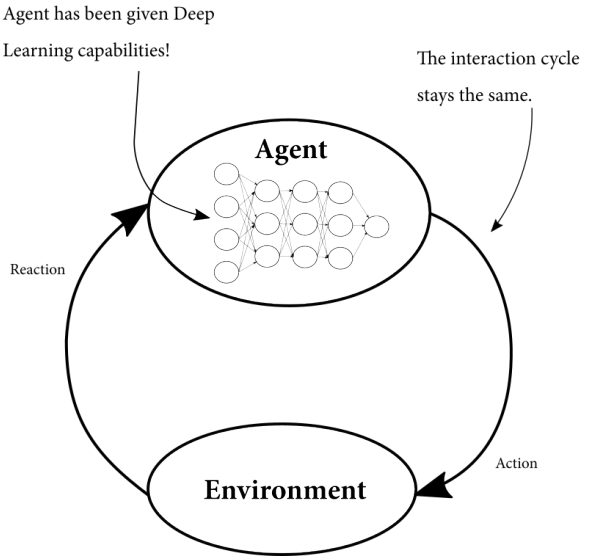
actions for the given state. At first, the agent selects random moves, but after several episodes, the agent learns to perform. I will cover how to solve this problem in more depth later chapters. For now, this should give you an idea how DRL works.

In the previous section, I mentioned how SL was able to deal with sampled feedback. That is, supervised approaches do not have to learn from all possible data, but instead, the task is to learn from a few samples and then generalize to unseen samples. Though not the only way to mix deep learning and reinforcement learning, the most popular use of deep reinforcement learning (DRL), is to use deep learning to approximate the state space of large reinforcement learning problems. By doing this, we can solve RL problems that were not possible to solve before.



We can now iterate over the reinforcement learning cycle we described earlier and simply give the agent deep learning capabilities to create the deep reinforcement learning cycle or API. Most of the time, the deep learning layer will be used to improve the agents' perception, but this is not the only way we can use deep learning to improve reinforcement learning agents as you will see through the chapters.

The deep reinforcement learning interaction cycle

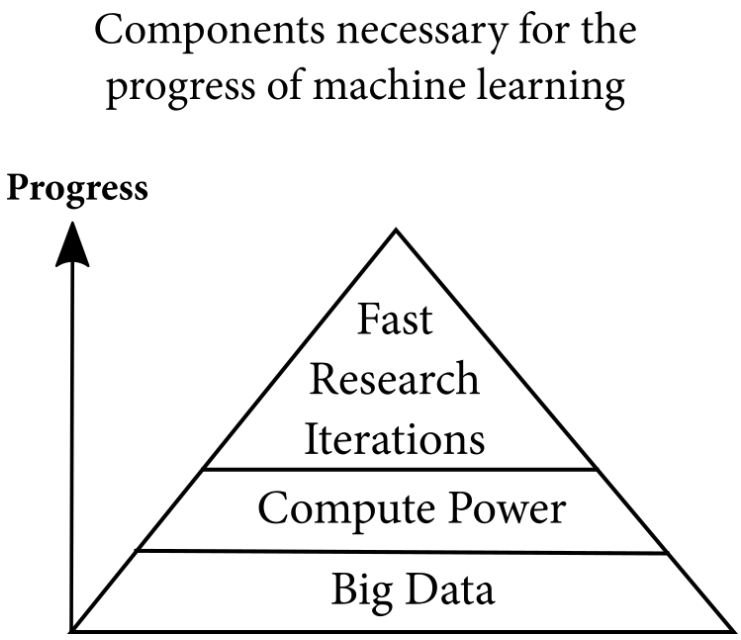


Why now?

You have come to the DRL party just in time. Deep reinforcement learning inherits a lot of history from both the deep learning and reinforcement learning fields. The events that are currently enabling DL research to advance allow DRL research to advance as well. The conditions are ideal for motivated individuals to start contributing, and current developments make the future of DRL bright.

The Learning Pyramid

The progress of machine learning research depends on three major conditions. First, it is vital to have *large amounts of quality data* to train ML models. Second, it is important to have *lots of compute power* available, because the process of training any ML model—especially the neural networks found in DL—is very computing intensive. Third, it is difficult to design and improve ML algorithms that train these models. The *ability to iterate rapidly* when creating a new algorithm has proven beneficial for the progress of the field. There three components is what I call the **learning pyramid**.



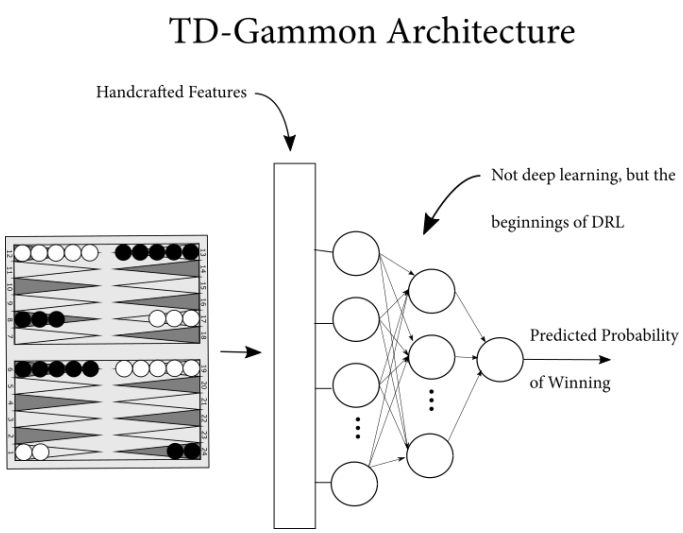
In the 1990's, there was a dramatic improvement to the first component of the learning pyramid with the invention of the internet. Lots of data is the oil that keeps the machine learning engine moving. It is only with lots of data that we can train deep models and make technological progress; do you remember how it was like before search engines? One of Google's strategic advantages over other search engines companies in the late 1990's was the collection and use of data to improve page rankings. Do you remeber AltaVista? Do you use Yahoo? Today, the use of deep learning models trained in large amounts of data is common. Don't believe me? Just ask Siri, Cortana, Bixby, Alexa or Google's Assistant. They might not only be able to tell you examples of DL applications, they are the examples. However, large amounts of data on their own are not sufficient. Training models on large datasets can still take days, weeks, and even months without enough computing power. But we have witnessed an exponential growth in the computing power available to us for decades, and this growth in the availability of computing power, particularly GPUs, that has enabled researchers to train neural network hundreds of times faster than before. In early 2010, graphics processing units (GPUs) were first used to train a neural network. GPUs are processing units like the CPUs regularly found on a computer. The main difference is that while CPUs are good at processing sequential data, GPUs are good at processing parallel data. When doing backpropagation in a deep NN, the calculation of the gradients can be parallelized and the speed up gains when doing so are large. For this reason, the processing of parallel data becomes a priority, and the availability of powerful GPUs a good thing for deep learning research.

Every year we continue to get more and more processing power, from both CPUs and GPUs. There are some that predict that this exponential growth will continue for years to come, possibly surpassing the computing power of a human brain in just a few decades. Some even suggest that we will surpass the processing power of all human brains combined in our lifetime. Think about that for a second!

Data and computing power form the basis for rapid iterative Machine Learning research because they allow researchers to iterate ideas quickly. You come up with an idea for a new algorithm, you code it up and send a few versions to the cloud, and you soon find yourself in the next iteration. Rapid development iterations allow researchers to come up with new and better algorithms much faster than they did before. Lately, we have been getting improvements and groundbreaking results from the deep learning community every month or so, and that by itself is amazing.

Success Stories

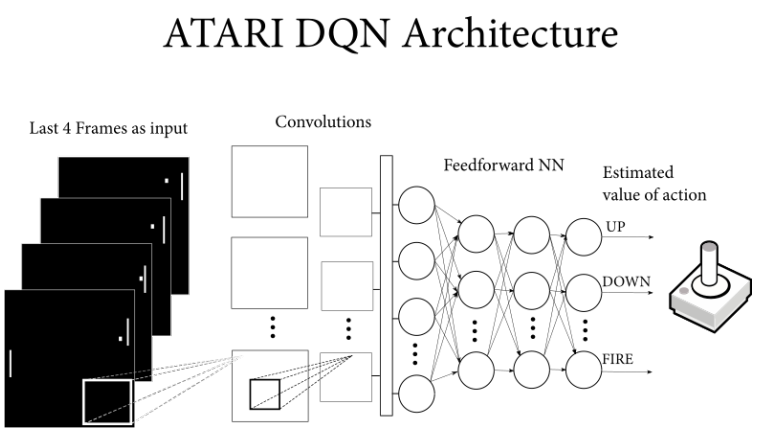
The use of artificial neural networks in reinforcement learning started around the 1990's. One of the well-known early RL successes was Gerald Tesauro's backgammon-playing computer program, called TD-Gammon. TD-Gammon learned to play backgammon by learning to evaluate table positions on its own through reinforcement learning. Back then, however, there was no possibility of training deep neural networks, so the best hand-crafted features were selected and passed to a regular network classifier instead. Even though the techniques implemented are not exactly considered deep reinforcement learning, TD-Gammon was one of the first widely-reported success stories of using NN to solve complex RL problems.



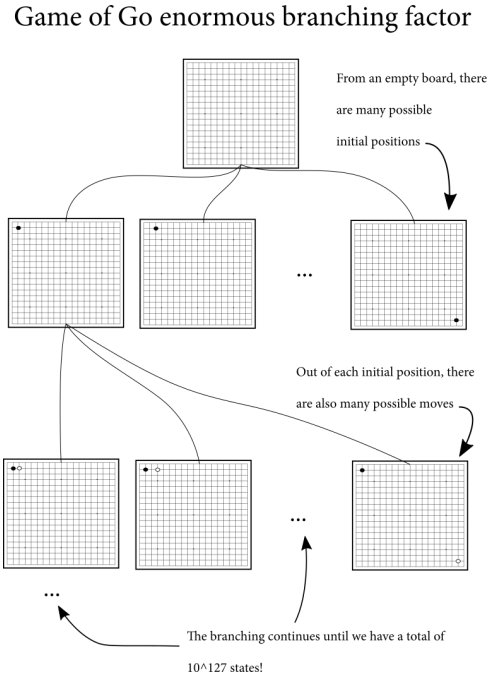
In 2004, Andrew Ng et al. developed an autonomous helicopter that teaches itself to fly stunts by observing hours of flights by experts. They used a technique known as **inverse reinforcement learning** (IRL), in which an agent uses expert demonstrations to derive the expert's goal. The same year, Kohl and Stone used a class of deep reinforcement learning methods known as **policy gradient** (PG) methods to develop a soccer playing robot for the RoboCup tournament. They used RL to teach the agent forward motion. After only three hours of training, the robot achieved the fastest forward moving speed of any other robot of the same hardware.

There were other successes in the 2000's, but the field of DRL really only started growing after the deep learning field took off in 2013, when DeepMind published a paper presenting the DQN algorithm. DeepMind trained an agent to play ATARI games straight from pixel images, using a single convolutional neural network and a single set of hyper-parameters. They trained the agent from scratch in almost 50

games. In 22 of the games, their algorithm reached better performance than a professional human player. This accomplishment started a revolution in the DRL community: Silver et al. released the DPG algorithm in 2014. Then, Lillicrap et al. iterated DPG with DDPG in 2015. In 2016, Schulman et al. released TRPO and GAE algorithms; Sergey Levine, Chelsea Finn, et al. released the GPS algorithm; and Silver et al. shocked the world with AlphaGo, following up with AlphaGo Zero and AlphaZero in 2017.



AlphaGo is a Go computer program that combines supervised learning, deep reinforcement learning, and Search. The program observed millions of professional players and then used reinforcement learning to perfect its game through self-play. It's important to note that the game of Go is one of the more complex board games in the world, reaching a state-space size of 10^{170}—more than the number of atoms in the universe, as I mentioned earlier. AlphaGo learned by bootstrapping on professional players; AlphaGo Zero, learned to play from scratch, only by its own trial-and-error learning, and it still beat AlphaGo 100 games to 0. AlphaZero,

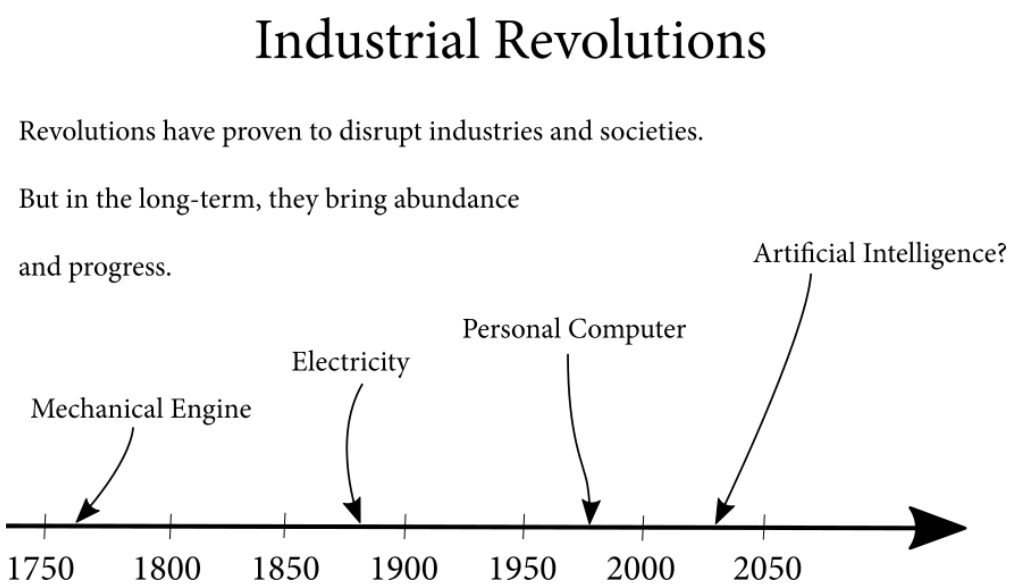


one of the biggest steps forward in deep reinforcement learning history, learned to play better than AlphaGo Zero, beating it 60 games to 40, and was trained in a shorter amount of time than all previous algorithms.

Thanks to deep reinforcement learning we've gone from training agents to play TD-Gammon, with its 10^{20} states, to training agents to play Go, with its 10^{170}, in just two decades. Imagine what the next two decades will bring us. Deep reinforcement learning is a booming field and is expected to revolutionize the artificial intelligence community.

Opportunities Ahead

Artificial intelligence is a field with unbounded potential for positive change regardless of what fearmongers say. Back in the 1750's, there was chaos due to the start of the industrial revolution. Machines were replacing repetitive manual labor and mercilessly displacing humans. Everybody was concerned; how could we become better than these machines and get our jobs back? What are we going to do now? The fact is these jobs were not only unfulfilling, but many of them were also dangerous. After 100 years of revolution, the long-term effects of these changes were benefitting communities. People that usually owned only a couple of shirts and a pair of pants were

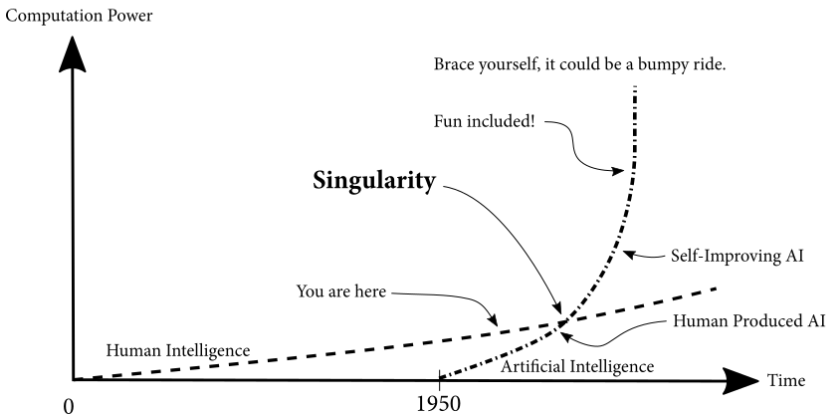


now able to get much more for a fraction of the cost. Surely, it was rough for society at first, but the long-term impact is the more important pursuit.

Artificial intelligence could be considered part of the last revolution called the digital revolution, but some proponents think AI is a revolution of its own. The digital revolution started in the 1970's with the introduction of the personal computers. Then the internet changed the way we do things. Because of the internet, we got big data and cloud computing. Machine learning used this fertile ground to sprout into what it is today. In the next couple of decades, the changes and impact to society will be difficult to accept at first, thus fearmongers. Sure, lots of people may lose their jobs as multiple industries change. Self-driving cars, autonomous drones, smart cities, speech recognition, personal assistants, virtual reality, just to name a few. As you can see, the long-lasting effect will be far superior to any setback along the way. I expect in some decades humans will no longer need to work for food, clothing or shelter as these things will be produced by AI continuously making our society thrive with abundance. How many people out there hate their jobs? Well, being freed from unsatisfying jobs should be one of the top humanity's goals for the century.

As we push the intelligence of machine to levels superior to that of ours, we will have access to creating intelligent machines we weren't capable before. This principle is called singularity; machines that are more intelligent than humans would allow for the improvement of intelligent machines at a faster pace, given that the self-improvement cycle would get rid of the bottleneck, namely, humans.

Singularity could be just a few decades away



But it is too soon to start dwelling on this. Singularity is part of the future and nobody knows with certainty when it will happen or even if it will happen. Thus, it is not worth spending too much time on it. If it can happen, it will but it is hard to tell when. Regardless, the best you and I can do is continue to work to advance artificial intelligence and forget about the rest. There is plenty of work to do, and that's why you are here.

When to use Deep Reinforcement Learning?

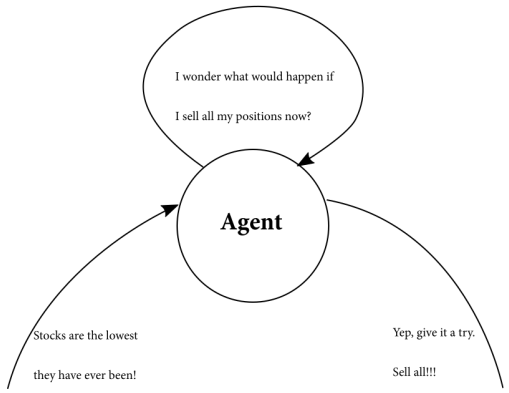
You could formulate any machine learning problem as a deep reinforcement learning problem, but this is not always a good idea, and for multiple reasons. You should know the pros and cons of using DRL in general, and you should be able to identify what deep reinforcement learning is good at and what it is not so good at, from a technological perspective.

What are the pros and cons?

Beyond a technological comparison, I would like you to think about the intuitive advantages and disadvantages of using deep reinforcement learning for your next project. You will see that each of the points highlighted can be either an advantage or a disadvantage depending on what kind of problem you are trying to solve. For example, this field is about letting the machine take control. Are you OK with letting the machine take the decisions for you?

There is a reason that deep reinforcement learning successes are often games: it could be very costly and dangerous to have an agent training directly in the real world. Can you imagine a self-driving car agent learning not to crash by crashing? The machine will have to make mistakes. Are you able to afford that? Are you willing to risk the negative consequences—actual harm—to humans? Whether you are writing agents for a home project or work, these questions should be considered beforehand.

Deep reinforcement learning agents will explore!
Can you afford mistakes?



You will also need to think about what exploration strategy to use. Deep reinforcement learning agents learn by trial-and-error. This means that they will have to select actions for the sake of gaining information. Such actions maybe terrible ideas, but this is how your agent learns. Unfortunately, the most commonly used exploration strategy today is called epsilon-greedy. It means that with an epsilon probability, a very small number, the agent will try a random action. The consequences could be disastrous in the real world. Give it a thought, what would happen with a stock trading agent that randomly explores the set of all possible actions? So, before you begin your quest to develop a trading bot, make sure you think about better ways for your agent to explore.

Finally, training from scratch every time can be daunting, time-consuming and resource intensive. However, there are a couple of areas that study how to bootstrap previously acquired knowledge. First, we have **transfer learning** which is about transferring knowledge gained in different tasks to new ones. For example, if you want to teach a robot to use a hammer and a screwdriver, you could reuse low-level actions learned on the "pick up the hammer" task and apply this knowledge to start learning the "pick up the screwdriver" task from a more advanced stage. This should make intuitive sense to you as humans don't have to relearn low-level motions each time they learn a new task. Humans seem to form hierarchies of actions as we learn. The field of **hierarchical reinforcement learning** tries to replicate this in deep reinforcement learning agents.

The second area that leverages previously acquired knowledge has to do with transferring knowledge learned from simulation into the real world. For example, training on a simulated robot and then running the learned policy on a real-world robot. For some problems, this is not a straightforward transfer, and there is some fine-tuning to be done. Keep an eye out for these topics in later chapters and think about leveraging these techniques in your projects.

What are Deep Reinforcement Learning's Strengths?

Deep reinforcement learning is about mastering explicit tasks. Unlike supervised learning, in which generalization is the goal (an ImageNet classifier can distinguish between 1,000 classes, and an RL agent can often only do one thing well even if that one thing is "playing multiple ATARI games"), reinforcement learning is better at concrete, well-specified tasks. If you can define well what the task is and use that as

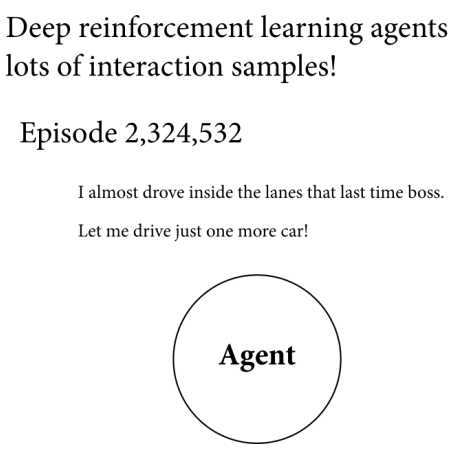
your reward function, you are more than halfway there. ATARI games, for example, have a very explicit goal. You even have the reward function that the agent should maximize shown explicitly on the screen. But sometimes the reward function is not clear. Deep reinforcement learning has difficulties when this is the case.

In deep reinforcement learning, we use generalization techniques to learn simple skills directly from raw sensory input. The performance of generalization techniques is one of the main improvements we've seen in recent years thanks to deep learning. These improvements have made deep reinforcement learning achieve stunning results in complex environments. If deep learning is good at some task, say image classification, then deep reinforcement learning problems that could leverage that will be partly solved.

Finally, another thing deep reinforcement learning has been good at for many years is learning from human demonstrations, also known as **apprenticeship learning** and **inverse reinforcement learning**. I talked briefly about the Stanford Helicopter that learned from watching the remote control moves by an expert, and we will touch on it in more detail later in the book.

What are Deep Reinforcement Learning's Weaknesses?

Of course, deep reinforcement learning is not perfect. One the biggest issues you will find is that agents need millions of samples to learn interesting policies. Humans, on the other hand, can learn from very few interactions. Sample efficiency is probably one of the top areas of deep reinforcement learning that could use some improvements. We will touch on this topic in several chapters as it is a very important topic in the field of DRL.



Another issue with deep reinforcement learning is understanding the meaning of rewards. If a human expert will be defining the rewards the agent is trying to maximize, does that mean that we are somewhat "supervising" this

agent? And more importantly, is this something we want to give up? We, as humans, don't have explicitly defined rewards. Often, the same person can see an event as positive or negative with only changing his or her perception of reality. What is a good reward? What is a bad reward? Additionally, rewards for a task such as walking are not very straightforward. Is it the forward motion that we are targeting, or is it not falling? There is ongoing research on reward signals. One very interesting research is called **intrinsic motivation**. Intrinsic motivation allows the agent to explore new actions just for the sake of it, out of curiosity. Agents that used intrinsic motivation showed an improved learning performance in environments with sparse rewards. I would expect more research to come out in this area in the future.

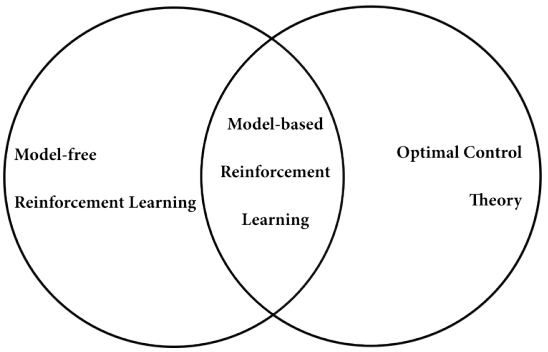
What will you learn in this book?

My goal is to take you, a machine learning enthusiast, from basic reinforcement learning techniques to state-of-the-art deep reinforcement learning. Therefore, we will focus on the reinforcement learning problem. You will first be shown the fundamental concepts in reinforcement learning. Then you'll build on those toward deep reinforcement learning techniques. In the last part of the book, I will walk you through the process of making deep reinforcement learning agents as general purpose as possible by touching on advanced topics related to artificial general intelligence.

Sequential Decision-Making

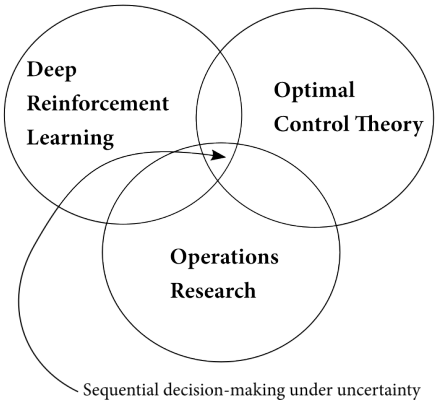
The reinforcement learning problem is about sequential decision-making under uncertainty. Sequential decision-making is a topic of interest to many fields, but each field looks at the problem through different lenses. Control theory studies ways to control complex known dynamical systems. Usually, we know mathematically how these systems behave before the interactions begin. Therefore, the methods and algorithms for solving control

Optimal Control Theory and Reinforcement Learning influence Model-Based Reinforcement Learning



theory problems make slightly different assumptions than deep reinforcement learning methods. However, the techniques used in control theory are often also used in deep reinforcement learning, and the other way around. Operations research also studies decision-making under uncertainty, but problems in this field often have much larger actions spaces than those commonly seen in deep reinforcement learning. Psychology studies human behavior, which is partly the same "sequential decision-making under uncertainty" problem. In fact, you can find research papers linking the different fields and proposing new theories of learning that all fields studying sequential decision-making are able to leverage. The bottom line is that you have come to a field that is influenced by a variety of others. Although this is a good thing, it also brings some inconsistencies in terminologies, notations and so on. I will use computer science notation and will let you know of different names you will find in the literature.

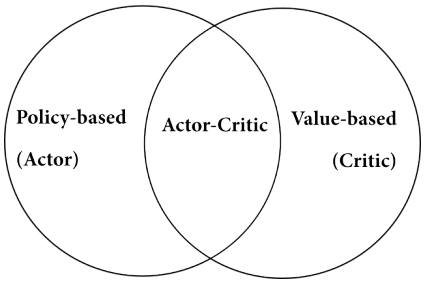
Many fields of science study decision-making. Probably the most important ones are

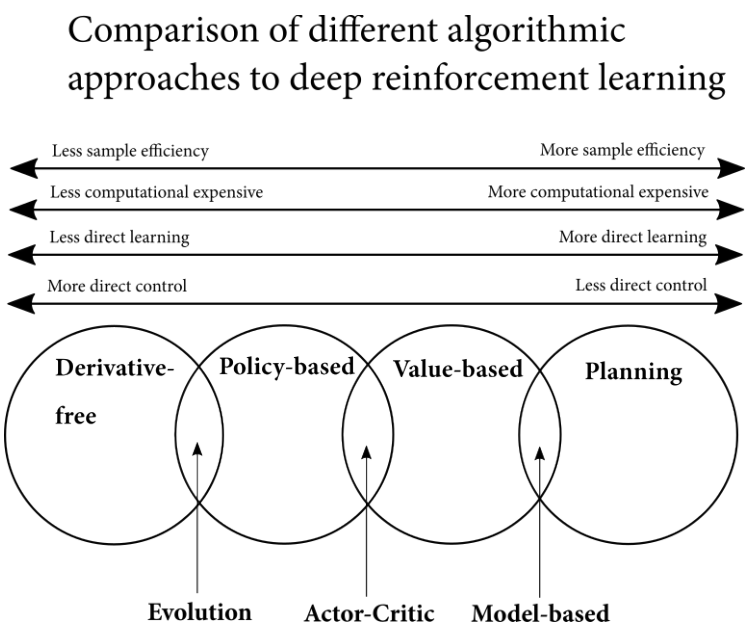


Algorithmic Approaches

Given all the different science fields studying sequential decision-making, the different approaches to solving reinforcement learning problems are many. In this book, you will learn most of them—planning methods, model-free prediction, model-free control, model-based reinforcement learning, and so on. Once we get to the "deep" reinforcement learning concepts, you will be presented with algorithms that are both value-based and policy-based, including some evolutionary approaches that are often not considered reinforcement learning techniques, despite solving reinforcement learning problems well. Then,

Hybrid policy and value based are often the more robust agents





you will be presented with the hybrid, actor-critic methods, followed by multi-agent reinforcement learning and concepts on artificial general intelligence, such as architectures, and safety.

I will take the hands-on approach to teaching these topics, starting with a very basic reinforcement learning environment that I will use to teach you different foundational concepts. Every chapter after that will start with a slightly more complex reinforcement learning problem, and I will present different ways to solve each one of these problems, introducing you to concepts, definitions, algorithm, code and yes, some math as well, as we progress.

Advanced Topics

In the last part of the book, I will address advanced deep reinforcement learning methods. First, I will go over advanced exploration strategies. The exploration vs. exploitation trade-off is unique and defining to reinforcement learning, so it is important we spend some time on it. After putting in the time, the realization of effects that different exploration strategies have on agent performance will come to manifestation.

Other advanced methods that you will discover in this book are hierarchical ap-

proaches, inverse reinforcement learning, and transfer learning. The goal of reinforcement learning is to solve intelligence in general, and this will become apparent as we advance. Therefore, you will uncover how some algorithms studied under the deep reinforcement learning umbrella also fall under the broader artificial general intelligence field. It's going to be fun.

What do you need?

Deep reinforcement learning is at the edge of artificial intelligence research, as such, you will need to bring a couple of things to the table.

Previous Knowledge

I presume you know what deep learning is beyond the refresher you just got in this chapter. Have you trained a convolutional neural network? Then great, if you haven't, please take some extra time to train a couple of networks. Trust me; you will get a lot more from this book if you do. However, you don't have to be a deep learning expert by any means. I'm going to teach deep reinforcement learning, and it is the reinforcement learning problems, concepts, and algorithms that sit at the core of this book. The deep learning networks used by deep reinforcement learning algorithms are often not nearly as complex as you see on the standalone deep learning field. I will show the deep learning architectures and comment briefly on them. However, you won't find detailed explanations as to why a network was used instead of a slightly different one. Most of this book will be on the application of these networks to deep reinforcement learning problems.

Ideally, you also have a basic machine learning knowledge. Are you familiar with supervised learning? Deep reinforcement learning at its core is the mix of reinforcement learning and supervised learning. It is important that you have a basic understanding of supervised learning techniques. The first chapter of the second part will give a refresher on supervised learning methods as related to RL, and this should be sufficient so long you already understand the basics.

Finally, as this will be a very hands-on take, make sure you feel comfortable with installing software, packages, and firing up your text editor of choice to code away problems. I will give you full implementations, but I hope that you type things along and experiment with them. It is amid the battle that heroes are born. My goal is to

unleash the best of you, so crack your knuckles, drink some coffee, and get ready.

Environment Setup

There are different challenges to consider when setting up and environment for DRL experimentation. One of them is the use of processing power. As expected, like deep learning techniques, GPUs are important to deep reinforcement learning. However, unlike deep learning, which does not use too much of the CPU, some deep reinforcement learning algorithms are asynchronous implementations that heavily rely on CPU processing power and most importantly thread count.

Another challenge is the need for graphical environments. You are probably familiar with using the cloud, headless servers, or Jupyter notebooks for training deep learning models. However, deep reinforcement learning methods have lots of interactions that can give you useful information just by looking at them. Though, it is still possible to send your code to the cloud, headless servers, or Jupyter notebooks, it is also an important debugging step to be able to see what your agents do during training. Have that in mind for when you start coding up your first couple of agents.

Finally, everyone has different needs, and it is difficult to provide a single environment that will satisfy all of the readers. Still, I will be providing a base environment setup for deep reinforcement learning development. However, feel free to customize your environment to better suit your needs. The provided environment is not designed to be better than any you could customize yourself. Invest the time, and you'll have fun galore.

A will to Learn (And be part of history)

I'm not trying to sound corny here; DRL gives you the opportunity to contribute to the world and be a part of history. To be part of history you don't necessarily need to be the best DRL researcher in the world, perhaps not even a "researcher", but you do need to be the best you. The motivation to learn—to read one more chapter—is all you'll need to get you through the difficult parts of this book. Learning is no different than working out; you may find challenges to get started, you may find it difficult to keep going, but the relentless drive to get through one more rep will bring the gains you desire. So, when you face one of those difficult lines, think about the long-term and push through.

Summary

By now, you should know a little bit of the history of artificial intelligence and that its goal is something that humans have pursued for many years. You should know that machine learning is one of the most popular and successful approaches to artificial intelligence. You should know that reinforcement learning is a one of the three branches of machine learning along with supervised learning and unsupervised learning. You should know that deep learning is not tied to any specific machine learning branch, but its power, has instead helped the entire machine learning community, including reinforcement learning, advance. You should know that deep reinforcement learning is simply the use of multiple layers of powerful function approximators known as neural networks to solve complex sequential decision-making problems. You should know that deep reinforcement learning has performed well in many control problems, but nevertheless, you should also have in mind that releasing human control for important decision making should not be taken lightly. You should know that some of the main issues in this field are the need for samples and the need for exploration. You should also have an idea of what is coming, the dangers of this technology, but more importantly you should be able to see the potential in this field and feel excited and compelled to bring your best and to embark in this journey. The opportunity of influence a change of this magnitude happen only every few generations. We should be glad it's us living these times. Let's be part of it.

More concretely, by now you:

- Understand what deep reinforcement learning is and how it began.
- Know how a larger field of related approaches share interests and concepts with deep reinforcement learning and how these relationships influence the field.
- Recognize why and how deep reinforcement learning is important and different than other approaches to machine learning.
- Can identify what deep reinforcement learning can do for different kinds of problems.